Statvfs()

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-17

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5709 bytes

<table>
<tr><td><b>Attack Category</b></td><td><ul><li>Path spoofing or confusion problem</li></ul></td></tr>
<tr><td><b>Vulnerability Category</b></td><td><ul><li>Indeterminate File/Path</li><li>TOCTOU - Time of Check, Time of Use</li></ul></td></tr>
<tr><td><b>Software Context</b></td><td><ul><li>File Management</li></ul></td></tr>
<tr><td><b>Location</b></td><td><ul><li>sys/statvfs.h</li></ul></td></tr>
<tr><td><b>Description</b></td><td><p>The statvfs() function returns information about a mounted file system which is specified by a named path.</p><p>statvfs() is a "check" type operation that may be associated with a time-of-check, time-of-use vulnerability. In other words, there is a risk that whatever information statvfs() reports may be invalid by the time any action is taken based on this information. This may pose a security loophole that an attacker could exploit.</p></td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>Function Name</th><th>Comments</th></tr><tr><td>statvfs</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td><p>The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.</p><p>An attacker may exploit the race condition between checking the file system using statvfs() and using this information by making a change to the file system between these two events. This may allow the attacker to subvert any security logic implicit in the use of statvfs() information.</p></td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

Solutions

Solution Applicability	**Solution Description**	**Solution Efficacy**
When statvfs() information is needed.	If the information reported by statvfs() is to be used in any way that may have security implications, then fstatfvs(), which uses a file descriptor instead of a named path, should be used instead. Subsequent operations should be performed using this file descriptor.	Likely to be effective in cases where subsequent "use" can be done using the same file descriptor. Efficacy depends on specifics of usage.
Generally applicable.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.
Generally applicable.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more

<table>
<tr><td></td><td><table>
<tr><td></td><td></td><td>difficult to exploit.</td></tr>
<tr><td>Generally applicable.</td><td>Limit the spread of time (cycles) between the check and use of a resource.</td><td>Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.</td></tr>
<tr><td>Generally applicable.</td><td>Recheck the resource after the use call to verify that the action was taken appropriately.</td><td>Effective in some cases.</td></tr>
</table></td></tr>
<tr><td><b>Signature Details</b></td><td>int statvfs(const char *restrict path, struct statvfs *restrict buf);</td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td>statvfs("somePath", &buf);
// operate on "somePath"</td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td>fstatvfs(fileDescriptor, &buf);
// operate on fileDescriptor</td></tr>
<tr><td><b>Source Reference</b></td><td>• http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/ucd-ecs-95-09.pdf[2]</td></tr>
<tr><td><b>Recommended Resource</b></td><td>• man page for statvfs()[3]</td></tr>
<tr><td><b>Discriminant Set</b></td><td><table>
<tr><td><b>Operating System</b></td><td>• UNIX</td></tr>
<tr><td><b>Languages</b></td><td>• C
• C++</td></tr>
</table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com